IdentityForge LDAP Gateway Setup and Configuration

By: Jamis Eichenauer

Last Updated: July 6, 2015

Contents

File preparation and downloads

Download the latest Java JDK from
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download the IdentityForge i5 Advanced Adapter Enterprise from
http://dl.empowerid.com/i5_advanced_adapter_5.3.0.0.3_bp03_enterprise.zip

Download JTOpen from
http://sourceforge.net/projects/jt400/

Download the Microsoft Visual C++ 2008 Redistributable Package (x86) from
http://www.microsoft.com/en-us/download/details.aspx?id=29

Download Win32 OpenSSL Light from
https://slproweb.com/products/Win32OpenSSL.html

Download ImportKey.class from
http://www.agentbob.info/agentbob/79-AB.html
(right click > Save As on the ImportKey.class link near the bottom of the page)
Alternate Link: http://www.emanuelis.eu/wp-content/uploads/2010/05/ImportKey.class.zip

Download JExplorer (an open source LDAP client) from
http://jxplorer.org/

Installation and configuration of the Java environment

Execute the Java JDK installer and click Next. On the optional features page, change "Public JRE" to "This feature will not be available". Change "Source Code" to "This feature will not be available". Click on "Development Tools" and then click the "Change.." button. Change "C:\Program Files (x86)\Java\jdk1.7.0_25" to a shorter path without spaces in it – for example: "C:\software\jdk1.7.0_25\". Click Next to begin the installation. Click Close when finished.

Click Start > Control Panel > System. Click on Advanced system settings. Click on Environment Variables. Under System variables, click on New…. For Variable name:, enter JAVA_HOME. For variable value:, enter the installation path of the Java JDK. For example: *C:\software\jdk1.7.0_25*

Installation of the IdentityForge LDAP Gateway

Extract i5_advanced_adapter_5.3.0.0.3_bp03_enterprise.zip to the root of the partition (C:\ in our example) and then navigate to C:\ i5_advanced_adapter_5.3.0.0.3_bp03_enterprise.zip\etc\LDAP Gateway. Extract ldapgateway.zip to the root of the partition (C:\ in our example).

Extract jtopen_7_10.zip to a folder named jtopen_7_10 in the root of the partition (C:\ in our example). Navigate to C:\jtopen_7_10\lib\ and copy jt400.jar and util400.jar to C:\ldapgateway\lib

Navigate to C:\ldapgateway\bin and open run.bat in Notepad. Adjust the following line to the installation path of the Java JDK:

```
set JAVA_HOME=\software\jdk1.6.0_16
```

In our example, this line would be changed to the following:

```
set JAVA_HOME=C:\software\jdk1.7.0_25
```

Configuration of the IdentityForge LDAP Gateway as400.properties file

Navigate to C:\ldapgateway\conf and open as400.properties in Notepad. Change the following parameter to true:

```
_isSSL_
```

Adjust the following parameters with the IP address of your target iSeries:

```
_host_
_agentHost_
```

Adjust the following parameters with the UID of the administrative AS/400 account:

```
_adminId_
_agentAdminId_
```

Place a # in front of the following parameters to comment them out:

```
_adminPwd_
_agentAdminPwd_
```

Remove the # from the following parameters to uncomment them:

```
_adminPwdEncrypt_
_agentAdminPwdEncrypt_
```

Clear text is not desirable for the passwords in the as400.properties file, so we will need to encrypt them using propertyEncrypt.bat. Copy C:\i5_Advanced_Adapter_5.0.0.4_Enterprise\scripts\propertyEncrypt.bat to C:\ldapgateway\dist and then open C:\ldapgateway\dist\propertyEncrypt.bat in Notepad. Adjust the following line to the installation path of the Java JDK:

```
set JAVA_HOME=C:\software\jdk1.5.0_15
```

In our example, this line would be changed to the following:

```
set JAVA_HOME=C:\software\jdk1.7.0_25
```

Scroll through the propertyEncrypt.bat file until you see the following line:

```
SET CLASSPATH=C:\software\identityforge\ldapgateway\dist\idfserver.jar
```

This path needs to point to the IdentityForge LDAP Gateway installation directory. In our example, this line would be changed to the following:

```
SET CLASSPATH=C:\ldapgateway\dist\idfserver.jar
```

Drop down to the end of the propertyEncrypt.bat file until you see the following lines:

```
rem Start Property Encrypt Utility
%JAVACMD%  %JVM_OPTS%  -cp %CLASSPATH%
com.identityforge.idfserver.util.AESCipherUtil idfRacfPwd
```

Change the bolded text to the password of the AS/400 administrative account, then save the propertyEncrypt.bat file. Double click propertyEncrypt.bat and you will see something similar to the following output:

```
New encrypted string as HEX: 10902AA71C4DF819C965E8B5B7DF0208
```

Copy this value (in our example, 10902AA71C4DF819C965E8B5B7DF0208) to the clipboard. Navigate to C:\ldapgateway\conf and open as400.properties in Notepad. Paste the value into the following parameters:

```
_adminPwdEncrypt_
_agentAdminPwdEncrypt_
```

Here is an example of a completed as400.properties file:

```
# USE SSL
_isSSL_=true

# HOST/IP VALUE TO CALL i5
_host_=74.125.225.114

# ADMIN ID TO CONNECT TO i5
_adminId_=AS400ADMIN

# ADMIN PASSWORD
#_adminPwd_=

# ADMIN ENCRYPTED PASSWORD
_adminPwdEncrypt_=10902AA71C4DF819C965E8B5B7DF0208

# HOST/IP WHERE AGENT RUNNING
_agentHost_=74.125.225.114

# ADMIN ID
_agentAdminId_=AS400ADMIN

# ADMIN PASSWORD
#_agentAdminPwd_=

# ADMIN ENCRYPTED PASSWORD
_agentAdminPwdEncrypt_=10902AA71C4DF819C965E8B5B7DF0208
```

Unpacking the IdentityForge LDAP Gateway environment

Navigate to C:\ldapgateway\dist and rename idfserver.jar to idfserver.zip. Extract the contents of idfserver.zip to C:\ldapgateway\dist\idfserver.

Configuring the Front-End LDAP Administrative Account

The next step is to set the Front-End LDAP administrative account and password. This is the account used by EmpowerID to bind to the IdentityForge LDAP Gateway. Open C:\ldapgateway\dist\idfserver\beans.xml in Notepad. Scroll down to the following section. We will be changing the Front-End LDAP administrative account listed in **bold**:

```
<bean name="as400" singleton="true"
class="com.identityforge.idfserver.backend.as400.As400Module">

<property name="suffix" value="dc=as400,dc=com"/>
<property name="workingDirectory" value="../as400"/>
<property name="adminUserDN" value="cn=idfAs400Admin, dc=as400,dc=com"/>
<property name="adminUserPassword" value="idfAs400Pwd"/>
<property name="altAdminUserDN" value="cn=oimAs400Admin, dc=as400,dc=com"/>
<property name="altAdminUserPassword" value="oimAs400Pwd"/>
<property name="allowAnonymous" value="false"/>
<property name="entryCacheSize" value="1000"/>
<property name="defaultUacc" value="read"/>
<property name="searchUsersType" value="user"/>
```

IdentityForge provides the ability to have two Front-End LDAP administrative accounts, but for our intents and purposes we only need one. Change the adminUserDN and altAdminUserDN property values to the DN of the Front-End account you wish to use to bind to LDAP. For example:

```
<bean name="as400" singleton="true"
class="com.identityforge.idfserver.backend.as400.As400Module">

<property name="suffix" value="dc=as400,dc=com"/>
<property name="workingDirectory" value="../as400"/>
<property name="adminUserDN" value="cn=EIDIDF, dc=as400,dc=com"/>
<property name="adminUserPassword" value="idfAs400Pwd"/>
<property name="altAdminUserDN" value="cn=EIDIDF, dc=as400,dc=com"/>
<property name="altAdminUserPassword" value="oimAs400Pwd"/>
<property name="allowAnonymous" value="false"/>
<property name="entryCacheSize" value="1000"/>
<property name="defaultUacc" value="read"/>
<property name="searchUsersType" value="user"/>
```

Now we need to change the password for the Front-End LDAP administrative account. Clear text is not desirable for the password in the beans.xml file, so we will need to encrypt it using propertyEncrypt.bat. Open C:\ldapgateway\dist\propertyEncrypt.bat in Notepad. Drop down to the end of the propertyEncrypt.bat file until you see the following lines:

```
rem Start Property Encrypt Utility
%JAVACMD%  %JVM_OPTS%  -cp %CLASSPATH%
com.identityforge.idfserver.util.AESCipherUtil idfRacfPwd
```

Change the bolded text to the password of the Front-End LDAP administrative account, then save the propertyEncrypt.bat file. Double click propertyEncrypt.bat and you will see something similar to the following output:

```
New encrypted string as HEX: 10902AA71C4DF819C965E8B5B7DF0208
```

Copy this value (in our example, 10902AA71C4DF819C965E8B5B7DF0208) to the clipboard. Change the adminUserPassword and altAdminUserPassword property values to the encrypted password string in the clipboard. For example:

```
<bean name="as400" singleton="true"
class="com.identityforge.idfserver.backend.as400.As400Module">

<property name="suffix" value="dc=as400,dc=com"/>
<property name="workingDirectory" value="../as400"/>
<property name="adminUserDN" value="cn=EIDIDF, dc=as400,dc=com"/>
<property name="adminUserPassword"
value="10902AA71C4DF819C965E8B5B7DF0208”/>
<property name="altAdminUserDN" value="cn=EIDIDF, dc=as400,dc=com"/>
<property name="altAdminUserPassword"
value="10902AA71C4DF819C965E8B5B7DF0208"/>
<property name="allowAnonymous" value="false"/>
<property name="entryCacheSize" value="1000"/>
<property name="defaultUacc" value="read"/>
<property name="searchUsersType" value="user"/>
```

Configuring the Back-End LDAP Administrative Account

The next step is to set the Back-End LDAP administrative password. This is the account used by EmpowerID to bind to the IdentityForge LDAP Gateway and sync inventory to the Back-End. Open C:\ldapgateway\dist\idfserver\beans.xml in Notepad. Scroll down to the following section. We will be changing the Back-End LDAP administrative account password listed in **bold**:

```
<bean name="hpbe2" singleton="true"
class="com.identityforge.idfserver.backend.hpbe.HPBEModule">
<property name="suffix" value="dc=system,dc=backend"/>
<property name="workingDirectory" value="../system"/>
<property name="schema" ref="schemas"/>
```

```
<property name="adminUserDN" value="cn=Directory Manager,
dc=system,dc=backend"/>
<property name="adminUserPassword" value="testpass"/>
<property name="altAdminUserDN" value="cn=Directory Manager,
dc=system,dc=backend"/>
<property name="altAdminUserPassword" value="testpass"/>
<property name="entryCacheSize" value="1000"/>
```

Clear text is not desirable for the password in the beans.xml file, so we will need to encrypt it using propertyEncrypt.bat. Open C:\ldapgateway\dist\propertyEncrypt.bat in Notepad. Drop down to the end of the propertyEncrypt.bat file until you see the following lines:

```
rem Start Property Encrypt Utility
%JAVACMD%  %JVM_OPTS%  -cp %CLASSPATH%
com.identityforge.idfserver.util.AESCipherUtil idfRacfPwd
```

Change the bolded text to the password of the Back-End LDAP administrative account, then save the propertyEncrypt.bat file. Double click propertyEncrypt.bat and you will see something similar to the following output:

```
New encrypted string as HEX: 10902AA71C4DF819C965E8B5B7DF0208
```

Copy this value (in our example, 10902AA71C4DF819C965E8B5B7DF0208) to the clipboard. Change the adminUserPassword and altAdminUserPassword property values to the encrypted password string in the clipboard. For example:

```
<bean name="hpbe2" singleton="true"
class="com.identityforge.idfserver.backend.hpbe.HPBEModule">
<property name="suffix" value="dc=system,dc=backend"/>
<property name="workingDirectory" value="../system"/>
<property name="schema" ref="schemas"/>
<property name="adminUserDN" value="cn=Directory Manager,
dc=system,dc=backend"/>
<property name="adminUserPassword"
value="10902AA71C4DF819C965E8B5B7DF0208"/>
<property name="altAdminUserDN" value="cn=Directory Manager,
dc=system,dc=backend"/>
<property name="altAdminUserPassword"
value=”10902AA71C4DF819C965E8B5B7DF0208"/>
<property name="entryCacheSize" value="1000"/>
```

Configuring SSL for the IdentityForge LDAP Gateway

To configure secure communications between EmpowerID and the IdentityForge LDAP Gateway we will need a PFX copy of the certificate being used by EmpowerID and an installed copy of OpenSSL to convert the PFX to a format usable by Java keystores.

If your EmpowerID certificate is not already available in PFX format, use the Certificates MMC snap-in to export the certificate and the corresponding private key to a PFX file. Be sure to note the pass phrase used to protect the PFX file; it will be needed later. If the certificate and its corresponding private key is already available in PFX format and the pass phrase for the PFX file is known, we can skip this step.

Next we need to install the needed conversion software. Install the Microsoft Visual C++ 2008 Redistributable Package (x86) by running vcredist_x86.exe and complete the installation. Next, install Win32 OpenSSL v1.0.1e Light by running Win32OpenSSL_Light-1_0_1e.exe. In our example, we are choosing an installation path of C:\software\OpenSSL-Win32. Click Next. When prompted on where to copy the OpenSSL DLLs to, choose "The OpenSSL binaries (/bin) directory". Click Next and finish the installation.

Copy the PFX certificate (named EIDcert.pfx in our example) to the root of the partition (C:\ in our example). Open a Command Prompt window and navigate to C:\software\OpenSSL-Win32\bin. Execute the following command:

```
openssl pkcs12 -in C:\EIDcert.pfx -out C:\EIDcert.pem
```

This will convert the PFX file to a PEM file. The OpenSSL toolkit will ask you to enter the import password; this is the pass phrase currently set on the PFX certificate. If you exported the certificate from the MMC snap-in, this will be the password you set on the certificate during the export. Enter the password for the certificate and press ENTER.

Next OpenSSL will prompt for the PEM pass phrase. We are setting a new pass phrase here – make sure to remember this value! You can use the same password as the import password if you want. Enter the pass phrase and press ENTER, then enter the pass phrase again to confirm and press ENTER. We now have a PEM file available at C:\EIDcert.pem.

Using a text editor such as NotePad++, open the PEM file (C:\EIDcert.pem in our example). We need to pull out the private key and put it into a separate file. Find the following lines below:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
DATA HERE
-----END ENCRYPTED PRIVATE KEY-----
```

Copy everything from the beginning of the -----BEGIN ENCRYPTED PRIVATE KEY----- line to the end of the -----END ENCRYPTED PRIVATE KEY----- line and paste them into a new text file named C:\encrypted.pem. Make sure there are no extra line breaks or spaces at the beginning or end of the file.

Next, we need to pull out the public key and put it into a separate file. Find the following lines below:

```
-----BEGIN CERTIFICATE-----
DATA HERE
-----END CERTIFICATE-----
```

Copy everything from the beginning of the -----BEGIN CERTIFICATE----- line to the end of the -----END CERTIFICATE----- line and paste them into a new text file named C:\cert.pem. Make sure there are no extra line breaks or spaces at the beginning or end of the file.

Next, we need to decrypt the RSA private key. Open a Command Prompt window and navigate to C:\software\OpenSSL-Win32\bin. Execute the following command:

```
openssl rsa -in C:\encrypted.pem -out C:\key.pem
```

When OpenSSL prompts you for the pass phrase, enter the PEM password you created earlier.

NOTE: Be sure to guard the unencrypted private key (C:\key.pem in our example) with extreme caution - loss of this file can result in compromise of your system. We recommend deleting this file after the next conversion.

Next, we need to convert the PEM files to DER files. Open a Command Prompt window and navigate to C:\software\OpenSSL-Win32\bin. Execute the following two commands:

```
openssl pkcs8 -topk8 -nocrypt -in C:\key.pem -inform PEM -out C:\key.der -outform DER
openssl x509 -in C:\cert.pem -inform PEM -out C:\cert.der -outform DER
```

Once these commands are completed, you will have two DER files. At this time it is recommended to delete the PFX and PEM files.

Next, copy ImportKey.class to the Java JDK bin folder (C:\software\jdk1.7.0_25\bin in our example). Open a Command Prompt and navigate to C:\software\jdk1.7.0_25\bin, then execute the following command:

```
java ImportKey C:\key.der C:\cert.der
```

This will merge the two DER certificate files into a single Java keystore. Note the following line: *"Using keystore-file : C:\Users\USERNAME\keystore.ImportKey".* Navigate to this directory and rename the keystore.ImportKey file to as400.jks and then copy it to the IdentityForge \conf directory (C:\ldapgateway\conf in our example)

The default password on the Java keystore is "importkey" without the quotes. If you want to change the Java keystore password, open a Command Prompt and navigate to C:\software\jdk1.7.0_25\bin, then execute the following command:

```
keytool -storepasswd -new NEWPASSWORDHERE -keystore C:\ldapgateway\conf\as400.jks
```

You will be prompted for the old Java keystore password, and then the new password will be set. Remember this password as it will be used later!

Finally, we need to configure the IdentityForge LDAP Gateway to point to this Java keystore. Open C:\ldapgateway\dist\idfserver\beans.xml in Notepad. Scroll down to the following section. We will be changing the Java keystore filename listed in **bold**:

```
<bean id="sslChannelFactory"
class="com.identityforge.idfserver.nio.ssl.SSLChannelFactory">
<constructor-arg><value>false</value></constructor-arg>
<constructor-arg><value>../conf/testnew.jks</value></constructor-arg>
<constructor-arg><value>abc123</value></constructor-arg>
<constructor-arg><value>false</value></constructor-arg>
</bean>
```

Change the second constructor-arg value to the name of the Java keystore we created earlier. For example:

```
<bean id="sslChannelFactory"
class="com.identityforge.idfserver.nio.ssl.SSLChannelFactory">
<constructor-arg><value>false</value></constructor-arg>
<constructor-arg><value>../conf/as400.jks</value></constructor-arg>
<constructor-arg><value>abc123</value></constructor-arg>
<constructor-arg><value>false</value></constructor-arg>
</bean>
```

The final step is to set the Java keystore password. We will be changing the Java keystore password listed in **bold**:

```
<bean id="sslChannelFactory"
class="com.identityforge.idfserver.nio.ssl.SSLChannelFactory">
<constructor-arg><value>false</value></constructor-arg>
<constructor-arg><value>../conf/as400.jks</value></constructor-arg>
<constructor-arg><value>abc123</value></constructor-arg>
<constructor-arg><value>false</value></constructor-arg>
</bean>
```

Clear text is not desirable for the password in the beans.xml file, so we will need to encrypt it using propertyEncrypt.bat. Open C:\ldapgateway\dist\propertyEncrypt.bat in Notepad. Drop down to the end of the propertyEncrypt.bat file until you see the following lines:

```
rem Start Property Encrypt Utility
%JAVACMD%  %JVM_OPTS%  -cp %CLASSPATH%
com.identityforge.idfserver.util.AESCipherUtil idfRacfPwd
```

Change the bolded text to the password of the Java keystore. The default password is “importkey” without the quotes – if you changed the keystore password earlier, please enter that value here instead. When finished, save the propertyEncrypt.bat file. Double click propertyEncrypt.bat and you will see something similar to the following output:

```
New encrypted string as HEX: 10902AA71C4DF819C965E8B5B7DF0208
```

Copy this value (in our example, 10902AA71C4DF819C965E8B5B7DF0208) to the clipboard. Change the Java keystore password listed in **bold** to the encrypted password string in the clipboard. For example:

```
<bean id="sslChannelFactory"
class="com.identityforge.idfserver.nio.ssl.SSLChannelFactory">
<constructor-arg><value>false</value></constructor-arg>
<constructor-arg><value>../conf/as400.jks</value></constructor-arg>
<constructor-arg><value>10902AA71C4DF819C965E8B5B7DF0208</value></constructor-
arg>
<constructor-arg><value>false</value></constructor-arg>
</bean>
```

Finally, since we are using an encrypted password for the Java keystore, we need to change the last constructor-arg value to true. For example:

```
<bean id="sslChannelFactory"
class="com.identityforge.idfserver.nio.ssl.SSLChannelFactory">
<constructor-arg><value>false</value></constructor-arg>
<constructor-arg><value>../conf/as400.jks</value></constructor-arg>
<constructor-arg><value>10902AA71C4DF819C965E8B5B7DF0208</value></constructor-
arg>
<constructor-arg><value>true</value></constructor-arg>
</bean>
```

Configuring SSL for the AS/400

To configure secure communications between the IdentityForge LDAP Gateway and the AS/400 we will need to fetch the SSL certificate from the OS/400 Certificate Manager. In a web browser, go to the Digital Certificate Manager on http://OS400domain:2001, where OS400domain is the OS/400 target system. Use the same user account and password that you use to access the OS/400.

In the left panel, select Create Certificate Authority. If the Create Certificate Authority is not an option, select Install Local CA Certificate on Your PC. Select Install Certificate and copy the certificate to a text file. In our example we are copying the cert to the root of the partition, to C:\cert.cer.

We now need to add the SSL certificate from the OS/400 system to the Java JDK keystore. Open a Command Prompt and navigate to the Java JDK bin folder. In our example, this is C:\software\jdk1.7.0_25\bin. Execute the following command:

```
keytool -importcert -file C:\cert.cer -alias arbitraryaliashere -keystore
C:\software\jdk1.7.0_25\jre\lib\security\cacerts
```

The alias is just a text string used to reference the certificate – you can enter any value. Enter the default JDK keystore password “changeit” without the quotes to confirm the addition.

To verify the presence of the certificate in the certificate store, run the following command:

```
keytool -list -keystore C:\software\jdk1.7.0_25\jre\lib\security\cacerts
```

Enter the defaukt JDK keystore password “changeit” without the quotes to view the contents of the Java keystore.

Allowing the IdentityForge LDAP Gateway and AS/400 to communicate through a firewall

The following ports may need to be opened between the IdentityForge LDAP Gateway and the AS/400:

Port 446 (TCP) – DDM
Port 448 (TCP) – Secure DDM
Port 449 (TCP) – Server mapper
Port 8470 (TCP) – Central server
Port 8475 (TCP) – Remote command and program call server
Port 8476 (TCP) – Signon server
Port 9470 (TCP) – Secure central server
Port 9475 (TCP) – Secure remote command/ Program call server
Port 9476 (TCP) – Secure signon server

For more information, please see the following JTOpen and IBM iSeries documentation pages:

Toolbox for Java and JTOpen
Port numbers for host servers and server mapper

Packaging the IdentityForge LDAP Gateway environment

Navigate to C:\ldapgateway\dist\idfserver, press CTRL+A to select all of files and folders in this location, then right click and choose Send to > Compressed (zipped) folder. This will create a .zip file in the C:\ldapgateway\dist\idfserver directory. Rename the .zip file to idfserver.jar. Copy idfserver.jar to C:\ldapgateway\dist. Overwrite the existing file.

Starting the IdentityForge LDAP Gateway

Execute C:\ldapgateway\bin\run.bat to start the IdentityForge LDAP Gateway.

Configuring the IdentityForge LDAP Gateway windows service wrapper

To host the IdentityForge LDAP Gateway java executable as a windows service, navigate to C:\ldapgateway\win_service and open IDF-Win-Server.bat in Notepad. Scroll down to the following section. We will be changing the JAVA_HOME and JVM path variables listed in bold:

set JAVA_HOME=**C:\Program Files\Java\jre7**
set JVM=**C:\Program Files\Java\jre7\bin\client\jvm.dll**

This path needs to point to the installation path of the Java JDK. In our example, this line would be changed to the following:

set JAVA_HOME=**C:\software\jdk1.7.0_25**
set JVM=**C:\software\jdk1.7.0_25\jre\bin\client\jvm.dll**

Scroll through the IDF-Win-Service.bat file until you see the HOME and APPLICATION_SERVICE_HOME variables listed in bold:

```
set HOME=C:\IdfService\ldapgateway
set APPLICATION_SERVICE_HOME=C:\IdfService\ldapgateway\win_service
```

This path needs to point to the IdentityForge installation directory. In our example, this line would be changed to the following:

```
set HOME=C:\ldapgateway
set APPLICATION_SERVICE_HOME=C:\ldapgateway\win_service
```

Scroll through the IDF-Win-Service.bat file until you see the SERVICE_NAME value listed in bold:

```
set SERVICE_NAME=IdentityForgeService
```

This variable can be changed to a name of your choosing - this will be the name of the Windows service as shown in Service Manager.

Scroll through the IDF-Win-Service.bat file until you see the CG_STDOUTPUT variable listed below:

```
set CG_STDOUTPUT=%CG_LOGPATH%\IDFServiceOut.log
```

In order to disable verbose logging, this line should be changed to the following:

```
REM -- set CG_STDOUTPUT=%CG_LOGPATH%\IDFServiceOut.log
```

Scroll through the IDF-Win-Service.bat file until you see the CG_DESCRIPTION and CG_DISPLAY_NAME values listed in bold:

```
Set CG_DESCRIPTION="Identity Forge Service for LDAP Gateway"
set CG_DISPLAY_NAME=IdentityForgeService
```

These variables can be changed as you see fit – the text will become the description and the display name of the Windows service as shown in Service Manager, respectively.

Open a Command Prompt window and navigate to C:\ldapgateway\win_service. Execute the following command:

```
IDF-Win-Service.bat install
```

If you wish to remove the service at a later date, execute the following command:

```
IDF-Win-Service.bat remove
```

Troubleshooting and monitoring the IdentityForge LDAP Gateway

To check and monitor the IdentityForge LDAP Gateway log files, look for the log files located in C:\ldapgateway\logs.

To enable Java debugging, open C:\ldapgateway\bin\run.bat in Notepad and scroll down to the following lines:

```
rem Start Ldap Gateway Server
%JAVACMD% %DEBUG% %JVM_OPTS% %SECURE% -cp %CLASSPATH%
com.identityforge.idfserver.Main %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Add **-Djavax.net.debug=all** directly after the %CLASSPATH% variable. Once completed, it should look like the following:

```
rem Start Ldap Gateway Server
%JAVACMD% %DEBUG% %JVM_OPTS% %SECURE% -cp %CLASSPATH% -Djavax.net.debug=all com.identityforge.idfserver.Main %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Execute C:\ldapgateway\bin\run.bat to start the IdentityForge LDAP Gateway with Java debugging enabled.

To increase the memory available to the Java JVM, open C:\ldapgateway\bin\run.bat in Notepad and scroll down to the following lines:

```
rem Start Ldap Gateway Server
%JAVACMD% %DEBUG% %JVM_OPTS% %SECURE% -cp %CLASSPATH% com.identityforge.idfserver.Main %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Add **-Xms512m -Xmx1024m** directly after the %CLASSPATH% variable. Once completed, it should look like the following:

```
rem Start Ldap Gateway Server
%JAVACMD% %DEBUG% %JVM_OPTS% %SECURE% -cp %CLASSPATH% -Xms512m -Xmx1024m com.identityforge.idfserver.Main %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Execute C:\ldapgateway\bin\run.bat to start the IdentityForge LDAP Gateway with the specified minimum and maximum memory available to the Java JVM.